EARTHQUAKE ENGINEERING AND STRUCTURAL DYNAMICS
Earthquake Engng. Struct. Dyn. **27**, 377–396 (1998)

NEW METHOD OF GENERATING SPECTRUM COMPATIBLE ACCELEROGRAMS USING NEURAL NETWORKS

JAMSHID GHABOUSSI*† AND CHU-CHIEH J. LIN‡
Department of Civil Engineering, University of Illinois at Urbana, Champaign, Urbana, IL 61801-2397, U.S.A.

SUMMARY

A new method is proposed for generating artificial earthquake accelerograms from response spectra. This method uses the learning capabilities of neural networks to developed the knowledge of the inverse mapping from the response spectra to earthquake accelerogram. In the proposed method the neural networks learn the inverse mapping directly from the actual recorded earthquake accelerograms and their response spectra. A two-stage approach is used. In the first stage, a replicator neural network is used as a data compression tool. The replicator neural network compresses the vector of the discrete Fourier spectra of the accelerograms to vectors of much smaller dimension. In the second stage, a multi-layer feed-forward neural network learns to relate the response spectrum to the compressed Fourier spectrum. A simple example is presented, in which only 30 accelerograms are used to train the two-stage neural networks. This example demonstrates how the method works and shows its potential. © 1998 John Wiley & Sons, Ltd.

KEY WORDS: earthquake engineering; structural dynamics; artificial accelerograms; neural networks

1. INTRODUCTION

Earthquake response spectra are often used in analysis and design of structures. In some cases, it is desirable to develop an artificial earthquake accelerogram, or select an existing recorded accelerogram, compatible with a given response spectrum. The need for developing accelerograms from response spectra is increasing, as more non-linear dynamic analyses are being performed. Methods for generating realistic accelerograms are likely to become increasingly important, since the future design codes may require more non-linear dynamic analyses. This study is also motivated by the significant increase in the number of accelerograms being recorded by the numerous instruments during the recent earthquakes. The increasing volume of the recorded accelerograms permit the development of novel approaches to systematic processing and utilization of the massive volume of the recorded accelerograms in structural design and analysis, including methods for generating accelerograms from response spectra. This has been the main motivation for the study reported in this paper.

If we consider determining the spectra from accelerograms as a *forward* or *direct* problem, then, determining the accelerograms from their spectra is an *inverse* problem. In the case of the Fourier spectra, the mapping is reversible and the inverse problem can be uniquely solved. Given a Fourier spectrum, we can perform an Inverse Fourier Transform to determine the accelerogram which gave rise to that Fourier spectrum. However, in the case of response spectrum, the mapping is not uniquely reversible. In computing the response spectrum from an accelerogram significant amount of information in the accelerogram gets lost,

* Correspondence to: Jamshid Ghaboussi, Department of Civil Engineering, University of Illinois, 3118 NCEL, 205 N. Mathews Avenue, Urbana, IL 61801-2397, U.S.A. E-mail: jghabous@uiuc.edu
† Professor
‡ Research Assistant

CCC 0098–8847/98/040377–20$17·50

Received 22 April 1997
Revised 12 August 1997

thus, making it mathematically impossible to uniquely retrieve the accelerogram. However, it may be possible to develop one or more accelerograms whose response spectra are close to a given response spectrum. Numerous studies have addressed the problem of generating spectrum compatible accelerograms. The publications on this subject are too numerous to cite. The following is a small sample of the works on spectrum compatible accelerograms: Housner and Jennings,[1] Shinozuka and Sato,[2] Tsai,[3] Saragoni and Hart,[4] Kaul,[5] Levy and Wilkinson,[6] Iyengar and Rao,[7] Wong and Trifunac,[8] Polhemus and Cakmak,[9] Khan,[10] Kimura and Izumi,[11] Spanos and Mignolet,[12] Collins *et al.*,[13] Haddon,[14] Sabetta and Pugliese.[15]

In this paper we use the learning capabilities of neural networks to develop a method for generating accelerograms from response spectra. The primary objective is to use actual recorded accelerograms to train neural networks to learn to associate the response spectra with the corresponding accelerograms and in this way to develop the capability of generating accelerograms from response spectra.

Unlike the mathematically formulated approaches to the same problem, the internal workings of the neural-network-based methods are not transparent to the user. Like most other new methods, the proposed methodology must be evaluated by examining its performance. The lack of transparency of the internal workings of the neural network has no effect on the performance and the utility of the neural-network-based methods. Inspite of the lack of transparency, the learning capabilities of neural networks are well understood and documented in the literature (for example, see Reference 16).

The proposed method consists of two stages. In the first stage, a replicator neural network is used to compress Fourier spectrum of the accelerograms. In the second stage, a multi-layer feed-forward neural network is used to relate the discretized response spectra to the compressed Fourier spectra. The proposed method is demonstrated by using 30 accelerograms to train the neural networks. The performance of the trained neural networks is evaluated by generating accelerograms for novel response spectra.

2. NEURAL NETWORKS

Neural networks are massively parallel computational models, inspired by the structure and operation of biological brains. The massively parallel structure of neural networks gives rise to their learning capabilities. It is the learning capabilities of neural networks which set them apart from other mathematically formulated methods, and allow the development of neural-network-based methods for certain mathematically intractable problems.

Neural networks are assemblages of interconnected artificial neurons, or nodes. Signals propagate along the connections and the strength of the transmitted signals depend on the numerical weights which are assigned to the connections. Each neuron receives signals along the incoming connection, performs some simple operations, such as calculating weighted sum of the incoming signals and calculating an activation function, and sends signals along its outgoing connections. The knowledge learned by a neural network is stored in its connection weights. The learning taking place, when a *learning method* is used to modify the connection weights in such a way that a given input pattern produces a given output pattern. The patterns used in training the neural network is called the training set. During the training, a neural network acquires the knowledge from the input–output pairs in the training set, and stores that knowledge in its connection weights.

In this study we have used Multi-Layer Feed-Forward neural networks (MLFF). In MLFF networks the artificial neurons are arranged in layers. The neurons in each layer are fully connected to all the nodes in the next layer. The outer layers are the input and output layers and the intermediate layers are the hidden layers. The flow of signals is from the input layer to the output layer. Given an input pattern, the signals are propagated through the layers of the neural network to produce an output. During the training, while the connection weights have not fully matured, the output of the neural network will differ from the desired output. The output error is back propagated through the neural network and the connection weights are modified according to a learning rule. A number of learning rules are available.[16] Modern research on neural

networks has developed many algorithms to speed up the rate of learning in networks. In this research, we have used a combined form of the Quick-Prop algorithm[17] and local adaptive learning rate algorithm.[18]

Feed-forward neural networks consist of a number of layers of artificial neurons or processing units: the input layer, the output layer and a number of hidden layers. Two hidden layers are sufficient in most applications and, if the problem can be modelled with one hidden layer, that is generally preferred. The number of processing units in the input and the output layers is dependent upon how the problem is formulated for neural network representation. The number of processing units in the hidden layers is a difficult part of the network architecture determination. The number of processing units in the hidden layers determines the capacity of a neural network, which in turn is related to the complexity of the underlying knowledge base in the training data. However, the degree of complexity of the problem cannot easily be quantified, and its relation to the size of the neural network is not very well understood at the present. It should also be pointed out that the representation problem does not have a unique solution; many different network architectures can produce similarly satisfactory results. Trial and error is one method of architecture determination in current use. Method of adaptive determination of network architecture, developed by Ghaboussi and co-workers, has played an important role in their previous studies of the applications of neural network,[19–23] as well as in this study. The method of adaptive architecture determination was initially proposed by Wu[24] and Wu and Ghaboussi[25] and was further developed and refined by Joghataie *et al.*[26]

Neural networks are ideal in dealing with problems which do not have unique and mathematically precise solutions. Mathematically formulated methods are precise and universally valid over all the possible ranges of the parameters. Unlike mathematically formulated methods, the neural-network-based methods are imprecision tolerant, and they can give reasonable results only over limited range of parameters. The range of the input parameters over which neural networks are expected to give reasonable results is the range of input parameters covered by the data in their training sets. In this sense, we can restate the problem as follows: from a given response spectrum, the proposed neural-network-based method is intended to generate one or more accelerograms, similar to the accelerograms with which it was trained. In this way, we can use the increasing number of the recorded accelerograms to enlarge the training data set of the neural networks and get increasingly more reasonable accelerograms.

In the remainder of this paper we will use new notation which was introduced recently[27,28] to present neural networks in a compact and symbolic way and to facilitate their discussion. The general form of the notation is,

$$\mathbf{F} = \mathbf{NN}(\{\text{input parameters}\} : \{\text{NN architecture}\}) \tag{1}$$

where, the symbol **NN** denotes the output of a MLFF neural network, and the notation indicates that the vector **F** is the output of the neural network. The first argument describes the input to the neural network, while the second argument field describes the neural network architecture, i.e. the number of processing units in the input layer, the hidden layers, and the output layer, respectively, and its training history.

3. THE PROPOSED METHODOLOGY

The objective of this study is to develop neural-network-based methodology capable of producing a reasonable artificial accelerogram which has a response spectrum close to a specified response spectrum used as input. Moreover, the accelerogram produced from a given response spectrum should either have characteristics similar to the group of accelerograms used in the training of the neural networks (the training set), or it should be one of the accelerograms in that group. In the latter case, the proposed methodology selects one of the earthquake accelerograms in its training set which has a response spectrum close to the specified input response spectrum. If the response spectra of none of the accelerograms in the training set matches the input response spectrum, then the proposed neural-network-based methodology will be able to synthesize an

artificial earthquake accelerogram from its training set, in such a way that the synthesized accelerogram will have characteristics similar to those in its training set.

The proposed methodology is based on developing a neural network which takes the discretized ordinates of the pseudo-velocity response spectra as input, and the output of the neural network produces the real and imaginary parts of the Fourier spectra of the generated earthquake accelerograms. Such a neural network will be trained with the response spectra and Fourier spectra of a number of actual earthquake accelerograms. Since in discretizing the response spectra and Fourier spectra a reasonable accuracy should be maintained, they are discretized with a large number of discrete values. Consequently, the resulting network will be very large and very difficult to train. To overcome this problem, we propose a two stage method. We can think of the neural network achieving two functions. First, it learns to relate the discretized response spectra to a compressed form of the Fourier spectra, and then, it learns to relate the compressed Fourier spectra to the actual Fourier spectra. These two functions have been separated and individual neural networks have been trained for each function.

3.1. Replicator neural network for data compression

The architecture of replicator neural networks consist of five layers with identical input and output layers and a much smaller number of processing units in the middle hidden layer. These neural networks are trained to replicate in their output layer the vector given at their input layer. In going from the input layer to the output layer the signals pass through small number of processing units in the middle hidden layer, thus, the activations of the middle hidden layer form an internal compressed representation of the input data. Replicator neural networks were first studied by Kohonen.[29] Later Ackley *et al.*[30] studied the replicator neural networks in the context of the 'encoder problem', and Cottrell *et al.*[31] developed multi-layer feed-forward replicator neural networks. Some recent theoretical studies[32,33] have clarified certain fundamental aspects of the operation of replicator neural networks. These studies indicate that the replicator neural networks perform a mapping from the n-dimensional input vector space to a unit cube in the k-dimensional vector space of middle hidden layer, where k is much smaller than n. These studies show that the middle hidden layer of the replicator neural networks produce optimal source codes.

The data compression of the Fourier transform of the accelerogram is accomplished by a replicator neural network, which learns to replicate the Fourier transforms. The input and output of the replicator neural networks represent the Fourier spectra; when given a Fourier spectrum at its input, it replicates it in its output. In doing so, the replicator neural network is also performing a data compression. As shown in Figure 1, the replicator neural network has a middle hidden layer with far fewer artificial neurons than in its input and output layers. In going from the input layer to the output layer, it channels the information through the few neurons at the middle hidden layer. Therefore, the activations of these neurons encode a compressed version of the Fourier spectra.

Figure 1 shows the replicator neural network developed in this study. In fact, this figure shows two separate replicator neural network. One neural network processes the real part of the Fourier transform while the second neural network processes the imaginary part of the Fourier spectra of the same earthquake accelerograms. The architecture of the two neural networks are the same. The procedure for training and using of these neural networks are also shown in Figure 1. First, the Fast Fourier Transform (FFT) of the earthquake accelerogram is computed, and the real and imaginary part of the FFT are given as input to the two separate neural networks. The input layers for each neural network has 2049 nodes, which is the number of discrete points in the FFTs. The output layers have the same number of nodes as the input layers. The neural network has three hidden layers. The middle hidden layer, which is the compression layer, has 20 nodes in each replicator neural network and the outer hidden layers each have 120 nodes. The output of the two replicator neural networks represent the real and imaginary parts of the FFT of the replicated accelerogram.

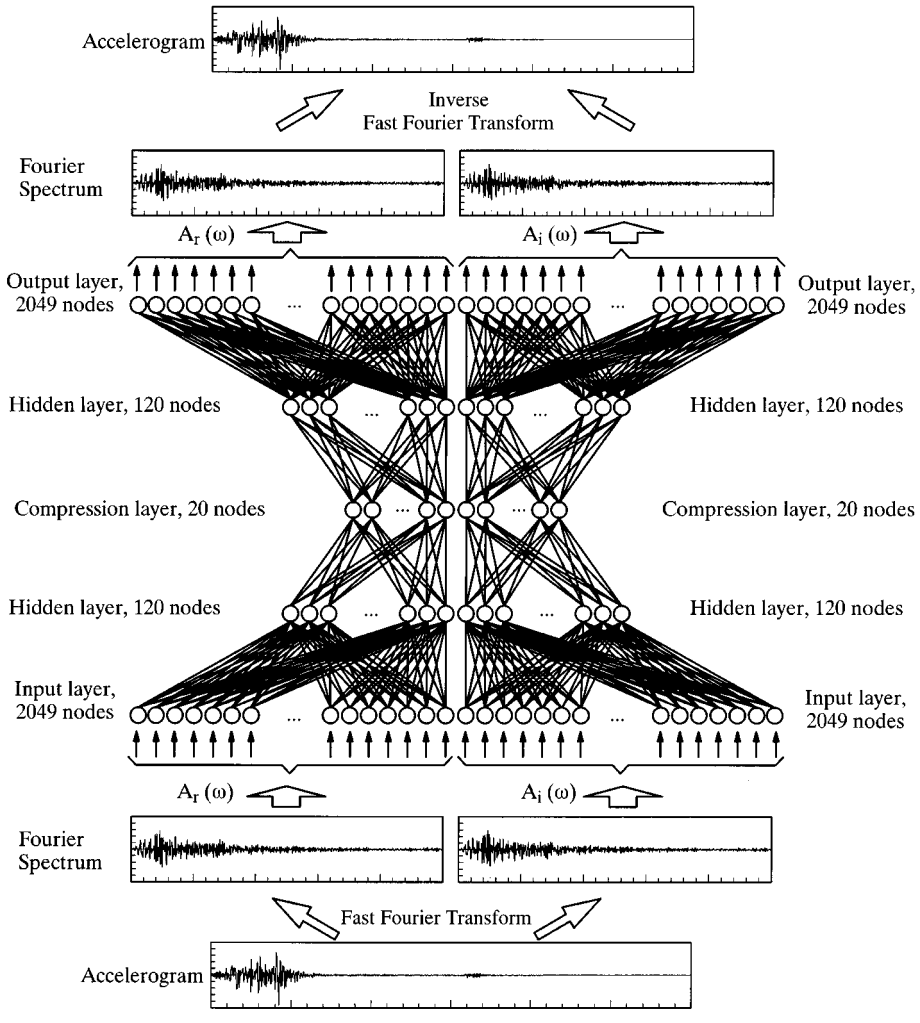

Figure 1. Replicator neural network used for data compression of the fourier transforms

It is useful to present the replicator neural networks in compact symbolic form. The input and output of the replicator neural networks represent the vectors of the real and imaginary parts of the Fourier transform, denoted by $\mathbf{A}_r$ and $\mathbf{A}_i$,

$$A(\omega) = A_r(\omega) + iA_i(\omega) = \int_0^\infty \exp(-i\omega t)\ddot{x}_g(t)\,dt \tag{2}$$

where $\ddot{x}_g(t)$ is the earthquake ground acceleration. The Fourier Transform has been computed by using the Fast Fourier Transform algorithm, as follows:

$$A(\omega) = A_r(\omega) + iA_i(\omega) = \sum_{t=0}^{N-1} \exp(-i2\pi\omega t/N)\ddot{x}_g(t)\Delta t \tag{3}$$

$$\mathbf{A}_r = \{A_r(\omega_j),\ j = 1, \ldots, 2049\} \tag{4}$$

$$\mathbf{A}_i = \{A_i(\omega_j),\ j = 1, \ldots, 2049\} \tag{5}$$

Table I. Earthquake accelerogram used in training neural networks

Date	Earthquake	Station	Magnitude*	PGA (g)	Duration [D]†
1940.5.18	El Centro	El Centro	7·0	0·43	29
1966.6.27	Parkfield	Cholame Shandon Array 2	6·1	0·49	12
1966.6.27	Parkfield	Cholame Shandon Array 5 355	6·1	0·36	8
1966.6.27	Parkfield	Cholame Shandon Array 5 85	6·1	0·43	7
1966.6.27	Parkfield	Temblor	6·1	0·39	4
1971.2.9	San Fernando	Lake Haghes Array 9	6·6	0·37	14
1971.2.9	San Fernando	Pacoima Dam 164	6·6	1·18	34
1971.2.9	San Fernando	Pacoima Dam 254	6·6	1·12	34
1972.11.23	Managua	Nicaragra 180	6·2	0·34	13
1972.11.23	Managua	Nicaragra 90	6·2	0·38	14
1979.8.6	Coyote Lake, CA	Gilroy Array #6 230	5·7	0·42	4
1979.10.15	Imperial Valley, CA	Bonds Corner	6·6	0·78	19
1979.10.15	Imperial Valley, CA	El Centro Array #2	6·6	0·39	13
1979.10.15	Imperial Valley, CA	El Centro Array #4	6·6	0·49	10
1979.10.15	Imperial Valley, CA	El Centro Array #5	6·6	0·51	36
1979.10.15	Imperial Valley, CA	El Centro Array #6	6·6	0·44	11
1979.10.15	Imperial Valley, CA	El Centro Array #7	6·6	0·46	10
1979.10.15	Imperial Valley, CA	El Centro Array #8	6·6	0·59	12
1979.10.15	Imperial Valley, CA	El Centro Differential Array	6·6	0·49	12
1979.10.15	Imperial Valley, CA	Casa Flores, Mexicali	6·6	0·43	16
1983.5.2	Coalinga		6·5	0·52	22
1983.7.12	Alaska		6·5	0·31	3
1983.7.22	Coalinga aftershock	CHP	5·9	0·54	6
1983.7.22	Coalinga aftershock	Oil City	6·0	0·85	10
1983.7.22	Coalinga aftershock	Pleasant Valley Pump Plant FF	6·0	0·41	4
1983.7.22	Coalinga aftershock	Pleasant Valley Pump Plant BS	6·0	0·44	3
1983.7.22	Coalinga aftershock	Pleasant Valley Pump Plant Yr	6·0	0·57	4
1983.7.22	Coalinga aftershock	Transmitter Hill 360	6·0	0·88	8
1983.7.22	Coalinga aftershock	Transmitter Hill 270	6·0	0·73	8
1983.11.16	Hawaii	Hilo-Fish and Wildlife	6·6	0·48	18

* The largest magnitude value of M_w, M_s, M_L, m_b, M_D
† 0·05 bracketed duration

The two replicator neural networks shown in Figure 1 are given by the following equations:

$$\mathbf{A}_r = \mathbf{NN}_r(\mathbf{A}_r\text{: } 2049,\ 120,\ 20,\ 120,\ 2049\} \tag{6}$$

$$\mathbf{A}_i = \mathbf{NN}_i(\mathbf{A}_i\text{: } 2049,\ 120,\ 20,\ 120,\ 2049\} \tag{7}$$

Each of these neural networks can be thought of as being composed of two neural networks; the lower three layers and the upper three layers.

The 30 earthquake accelerograms used in the training of the replicator neural networks are listed in Table I. The training of these very large neural networks was a lengthy process and consumed enormous amounts of computer time, about one week on a HP175/75 workstation. Much lower output errors on the training set were achieved for the replicator neural network for the real part of the FFT than for the imaginary part. This may be an indication that the imaginary part of the Fourier spectrum contains more complex information than the real part, thus, making the learning of the imaginary part more difficult.

The trained replicator neural networks were then tested by presenting accelerograms as input and comparing them with the neural network replicated accelerograms. These comparisons were first performed

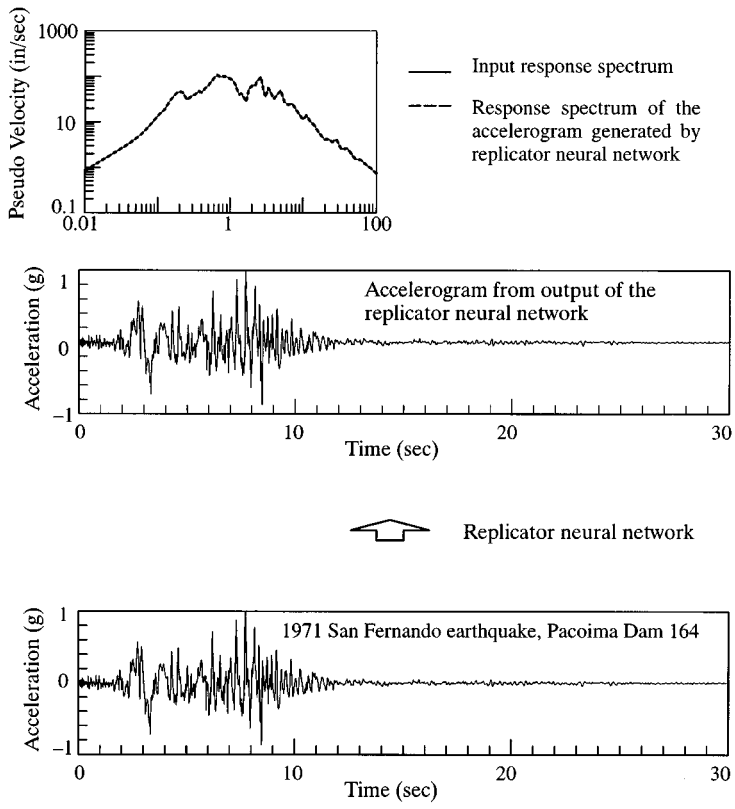

Figure 2. An example test of the trained replicator neural networks with an accelerogram from the training set

for the accelerograms in the training set listed in Table I, and next for novel accelerograms which were not included in the training set. A typical comparison for one of the accelerograms from the training set is shown in Figure 2. A similar comparison is shown in Figure 3 for a novel accelerogram. It can be seen that the replicator neural networks have learnt to generate an accelerogram which is very close to the input accelerogram and that the response spectrum of the generated accelerogram is very close to the response spectrum of the input accelerograms.

Each half of the replicator neural network can be considered an independent neural network performing a special function. The lower part of the replicator neural network (input layer to the middle hidden layer) performs encoding or data compression, while the upper part (middle hidden layer to the output layer) performs decoding or data decompression. The middle hidden layer of each replicator neural network can be considered as the output layer of the lower neural network and the input layer of the upper neural network. The activations of the middle hidden layer, which are the compressed Fourier spectra, are denoted by $\mathbf{A}_{rc}$ and $\mathbf{A}_{ic}$. Then the replicator neural networks of the equations (5) and (6) can be written as follows:

$$\mathbf{A}_{rc} = \mathbf{NN}_{rl}(\mathbf{A}_r\text{: } 2049, 120, 20) \tag{8}$$

$$\mathbf{A}_{ic} = \mathbf{NN}_{il}(\mathbf{A}_i\text{: } 2049, 120, 20) \tag{9}$$

$$\mathbf{A}_r = \mathbf{NN}_{ru}(\mathbf{A}_{rc}\text{: } 20, 120, 2049) \tag{10}$$

$$\mathbf{A}_i = \mathbf{NN}_{iu}(\mathbf{A}_{ic}\text{: } 20, 120, 2049) \tag{11}$$

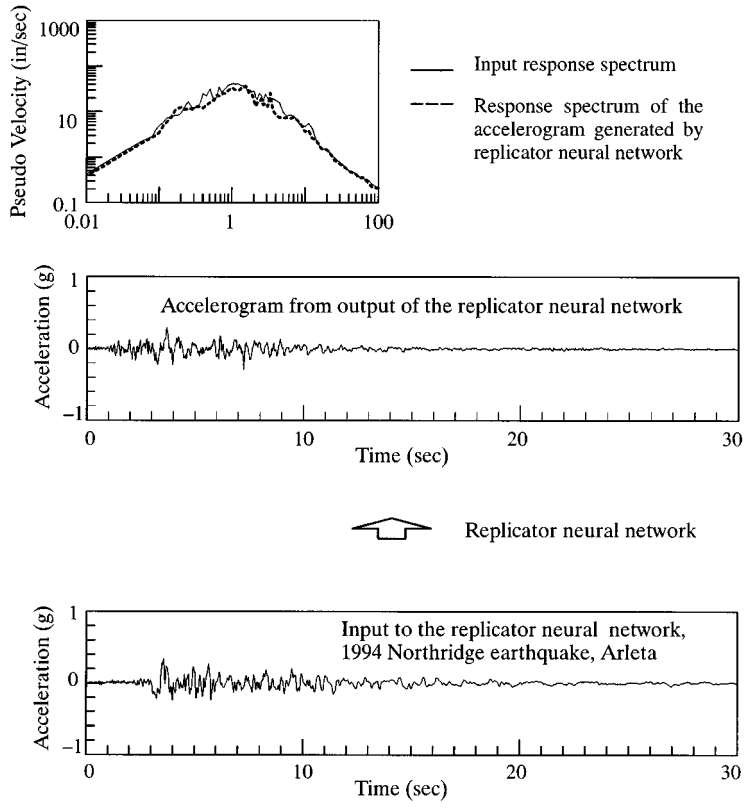

Figure 3. An example test of the trained replicator neural networks with a novel accelerogram from the test set

It is the upper part of the trained replicator neural network, $\mathbf{NN}_{ru}$ and $\mathbf{NN}_{iu}$ which is used in the Accelerogram Generator Neural Network.

3.2. *Accelerogram generator neural network*

The Accelerogram Generator Neural Network (AGNN) is developed to learn to generate the real and imaginary parts of the FFT of accelerogram from the vector of the pseudo-velocity response spectrum at discrete frequencies. The AGNN is shown in Figure 4. It can be seen the it is composed of two sections. The upper part of the AGNN is the upper-half of the trained replicator neural networks. The connection weights of upper section of the AGNN remain unchanged during the training of the rest of the neural network. The lower section is a neural network which relates the pseudo-velocity response spectrum to the compressed FFT. This part of the neural network has four layers. The input layer has 90 nodes which receive the values of the pseudo-velocity response spectrum at 90 discrete frequencies:

$$\mathbf{S}_v = \{S_v(\omega_j),\ j = 1, \ldots, 90\} \tag{12}$$

$$S_v(\omega) = \omega \max_t |x(t)| \tag{13}$$

$$\ddot{x}(t) + 2\xi\omega\dot{x}(t) + \omega^2 x(t) = -\ddot{x}_g(t) \tag{14}$$

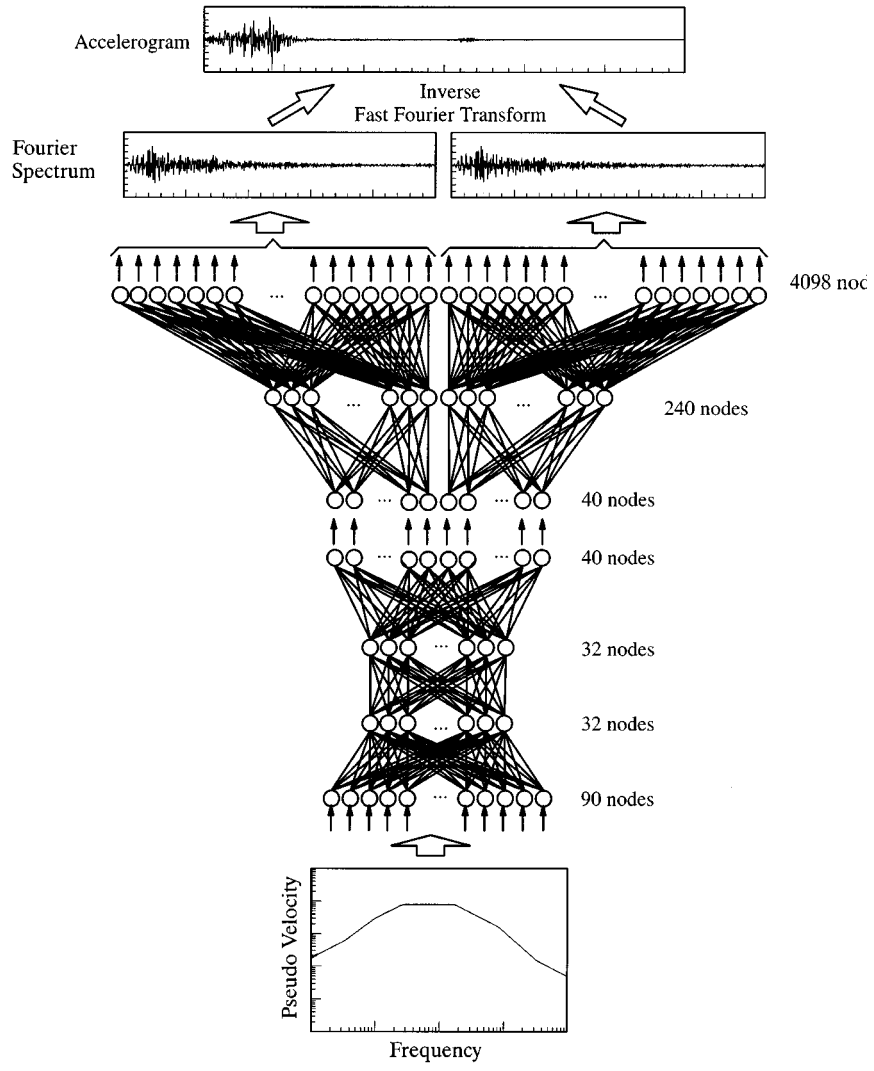

Figure 4. The accelerogram generator neural network

The output layer has 40 nodes representing the compressed FFT. The two hidden layer each have 32 nodes. The trained AGNN is given by the following equation:

$$\{\mathbf{A}_r, \mathbf{A}_i\} = \mathbf{NN}_{AGNN}(\mathbf{S}_v\colon 90, 32, 32, 40, 240, 4098) \tag{15}$$

As shown in the Figure 4, the Accelerogram Generator Neural Network is composed of three subneural networks. The lower four layers constitute a neural network, denoted by $\mathbf{NN}_s$, which relates the pseudo-velocity spectrum to the Compressed Fast Fourier Transform, and the two neural networks $\mathbf{NN}_{ru}$ and $\mathbf{NN}_{iu}$ form the upper part AGNN. The following equation symbolically shows the internal structure of the AGNN:

$$\{\mathbf{A}_r, \mathbf{A}_i\}\,\mathbf{NN}_{AGNN}(\mathbf{S}_v\colon\ ) = (\mathbf{A}_r\mathbf{NN}_{ru} + \mathbf{A}_i\mathbf{NN}_{iu})(\{\mathbf{A}_{rc},\ \mathbf{A}_{ic}\}\mathbf{NN}_s(\mathbf{S}_v\colon\ )\colon\ ) \tag{16}$$

The number of nodes in the hidden layers on $\mathbf{NN}_s$ were determined during the training of the neural network by using the adaptive architecture determination method.[26] The AGNN was also trained on the 30 earthquake accelerograms listed in Table I.

4. AN ILLUSTRATIVE EXAMPLE

The methodology proposed has been applied to a small data sample consisting of 30 recorded earthquake accelerograms for training (training data set) of the neural networks and 10 recorded earthquake accelerograms for testing of the trained neural networks (test data set). The earthquake accelerograms in the training data set are listed in Table I. These accelerograms are from earthquakes with magnitudes ranging from 5·6 to 7·6, and peak ground accelerations ranging from $0{\cdot}20g$ to $1{\cdot}47g$. The majority of these accelerograms have short durations, with $0{\cdot}05g$ bracketed durations of less than 20 s. Only eight accelerograms have a duration of between 30 and 40 s. No systematic criteria were used in selecting the accelerograms for this preliminary study, other than a lower bound on the magnitude. In future studies the accelerograms can be grouped according to source characteristics, site characteristics, duration and distance from source, and different neural networks can be developed and trained for each group.

All the accelerograms were discretized at 0·02 s and the durations of the strong shaking were variable. An arbitrary duration of 60 s was chosen for all the accelerograms and sufficient points with zero amplitude were added at the end of each accelerograms to bring the total durations of all the accelerograms to 60 s or 3001 discrete points. For the purpose of computing the Fast Fourier Transforms of the earthquake accelerograms, they had to be extended to $4096(2^{12})$, again by adding zero amplitude points to the end of the accelerograms. This resulted in the values of real and imaginary parts of the Fourier spectra at 4098 discrete frequencies. Because of the symmetry about zero frequency, only 2049 values of each of the real and imaginary parts were used. Since complete reversibility is essential, all the 2049 points were used as the input and output of the replicator neural networks. It is important that the full accelerogram be recoverable from its Fourier transform through Inverse Fast Fourier Transform.

All the pseudo-velocity response spectra were computed by using the Newmark Beta Method with $\beta = 0{\cdot}25$ and 2 per cent damping. The values of the response spectra were computed at 90 discrete frequencies equally spaced (in log scale) within the range of 0·01 and 100 Hz.

The two replicator neural networks, shown in Figure 1, were trained with the FFTs of the 30 accelerograms in the training data set. The architecture of each of the two replicator neural networks is described as the set $\{2049, 120, 20, 120, 1029\}$ which is the list of the number of the nodes in each layer. The upper-half of the trained replicator neural networks formed the upper portion of the Accelerogram Generator Neural Network, which relates the pseudo-velocity response spectra to the Fourier spectra. Only the lower portion of the AGNN was trained with the response spectra and the Fourier spectra of the 30 earthquake accelerograms in the training set, and the connection weights of the upper halves of the replicator neural networks remained unchanged during the training of AGNN. The final trained AGNN, which is shown in Figure 4, has the architecture $\{90, 32, 32, 40, 240, 4098\}$.

4.1. Testing of the trained accelerogram generator neural network

The trained neural network was tested with the earthquake accelerograms from both the training set and the novel cases from the test set. Figures 5–7 show the performance of the trained AGNN on three of the 30 earthquake accelerograms from the training set. For example, Figure 5 shows the time history and the response spectrum of 1971 San Fernando earthquake, Pacoima dam record at the bottom of the figure. The response spectrum for this accelerogram was provided as the input to the AGNN and its output provided the real and imaginary part of the Fourier Transform of the generated accelerogram. The upper portion of Figure 5 shows the generated accelerogram and its response spectrum. Comparison of the input and output accelerograms and their response spectra clearly indicates that the trained neural network has learnt the

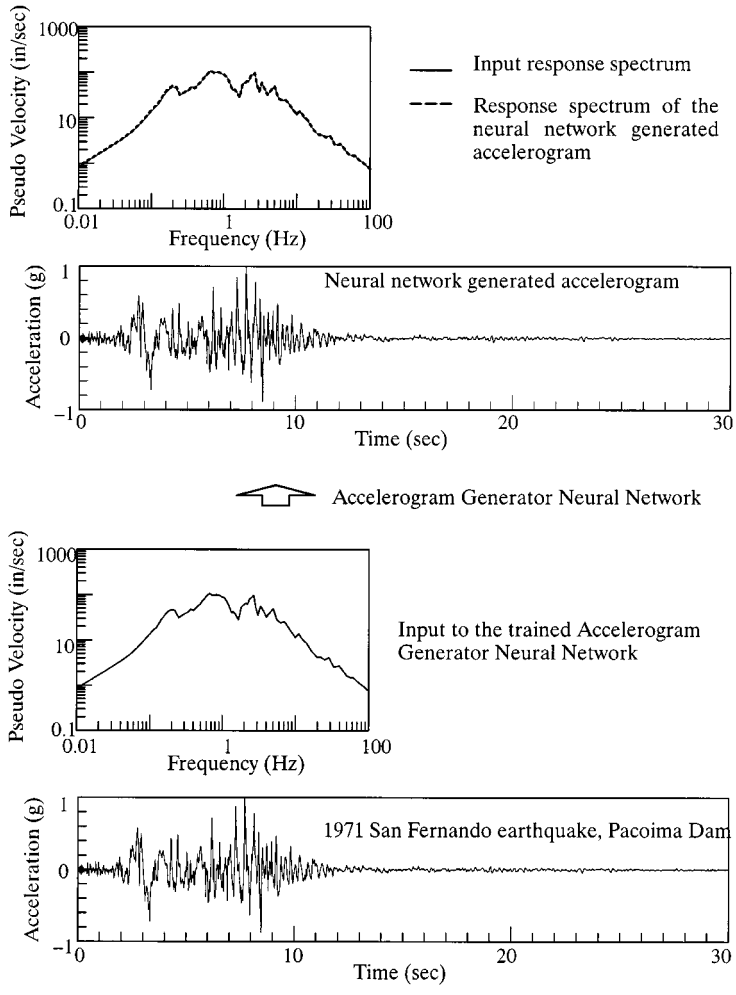

Figure 5. Test of the trained accelerogram generator neural network with data from its training set, 1971 San Fernando earthquake, Pacoima dam

training cases very well. Figures 6 and 7 show similar comparisons for 1983 Coalinga earthquake and 1983 Hawaii earthquake. The three earthquakes in Figures 5–7, obviously have different characteristics and the trained neural network performs equally well in generating these earthquakes from their response spectra.

When novel cases, which were not included in the training set, are used, the neural network may perform in two different ways. When given a novel response spectrum as the input, the neural network picks an accelerogram similar to one from its training set, if its response spectrum is close to the input response spectrum. Figures 8 and 9 show two such cases. In Figure 8, the input to the neural network is the response spectrum of Saratoga, Aloha Ave record from 1989 Loma Prieta earthquake, and the upper part of the figure shows that the neural network generates an accelerogram very similar to 1979 Imperial Valley earthquake, El Centro array #5 accelerogram. Similarly, Figure 9 shows that when the response spectrum for Sylmar record from 1994 Northridge earthquake in given as input, the neural network generates an accelerogram which is similar to the Pacoima dam accelerogram from the 1971 San Fernando earthquake. In both cases, it can be seen that the neural network has produced an earthquake record very similar to one from its training case, one which has a response spectrum very close to the input response spectrum.

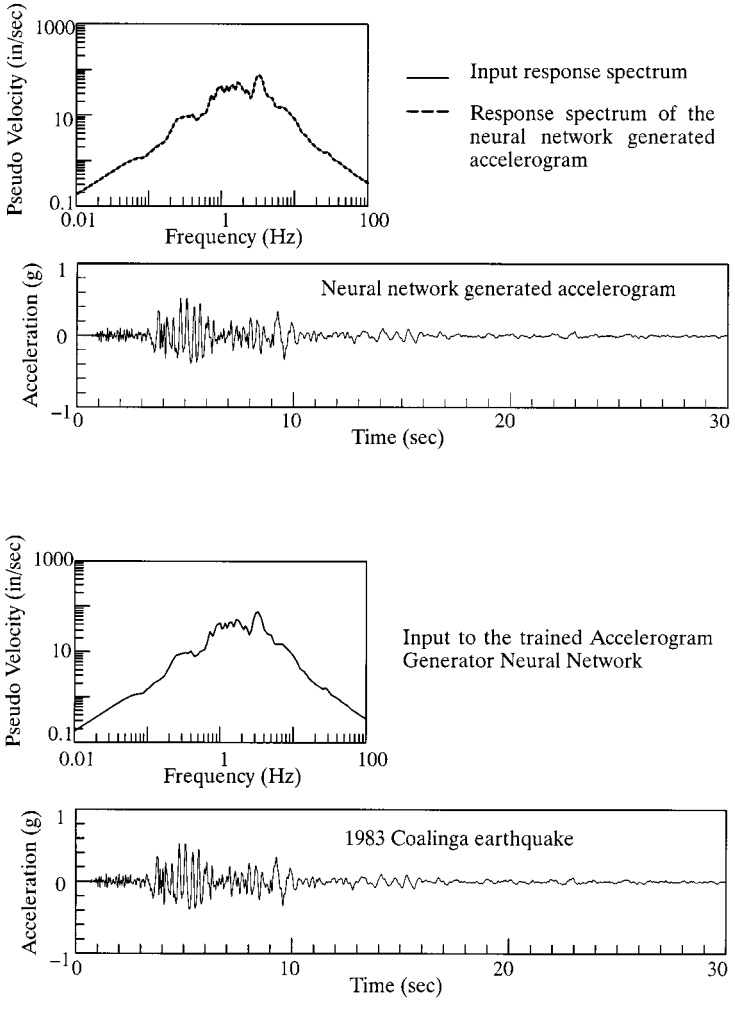

Figure 6. Test of the trained accelerogram generator neural network with data from its training set, 1983 Coalinga earthquake

The examples shown in Figures 8 and 9 demonstrate that the neural network generates a accelerogram very similar to the one from its training set which has the closest response spectrum to the input response spectrum. Next, we quantify this observation by introducing the error E_{jk} as a quantitative measure of the closeness of response spectrum $S_v(\omega)_k$ to response spectrum $S_v(\omega)_j$.

$$E_{jk} = \frac{\int |S_v(\omega)_j - S_v(\omega)_k| \, d\omega}{\int S_v(\omega)_j \, d\omega} \approx \frac{\sum |S_v(\omega_n)_j - S_v(\omega_n)_k| \, \Delta\omega_n}{\sum S_v(\omega_n)_j \, \Delta\omega_n} \tag{17}$$

The error E is computed from the logarithmic values of the response spectra and the integrals are evaluated numerically over the range of frequencies from 0·1 to 10 Hz. Table II presents the values of this error for the two cases shown in Figures 8 and 9. The first two columns presents the results for the case when the input to the neural network is the response spectrum of Saratoga, Aloha Ave record from 1989 Loma Prieta earthquake. The first column shows the error between the input response spectrum and the response spectra

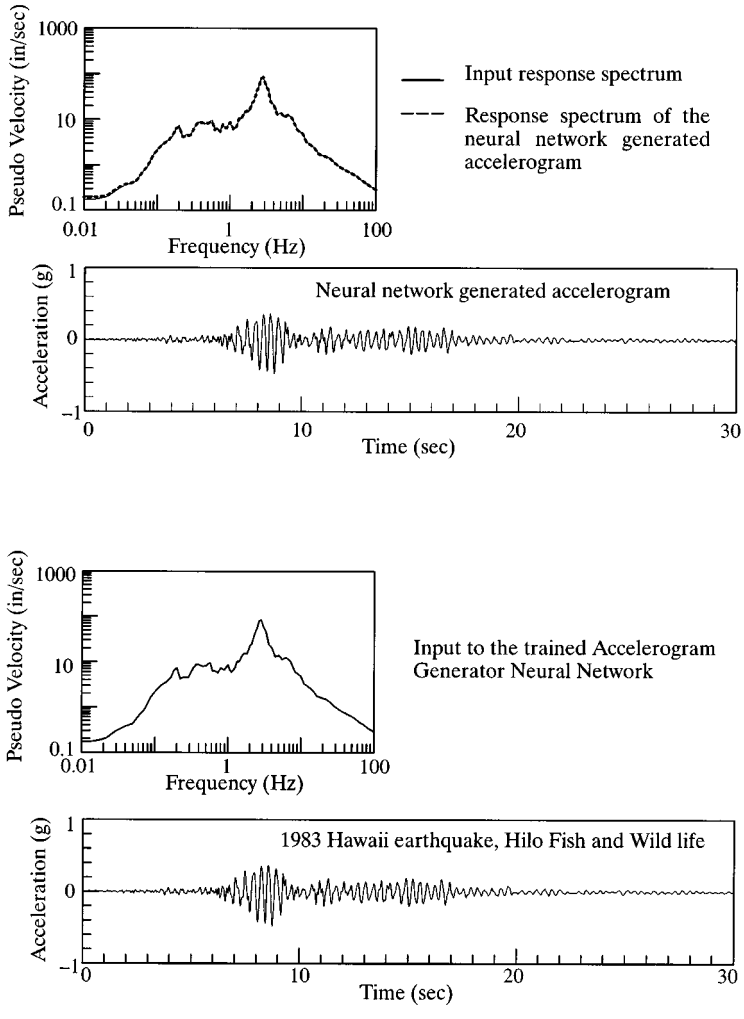

Figure 7. Test of the trained accelerogram generator neural network with data from its training set 1983 Hawaii earthquake, Hilo Fish and Wildlife

of the 30 accelerograms in the training set. The smallest entry is for the accelerogram number 15 which is the 1979 Imperial Valley earthquake, El Centro array #5 accelerogram which indicates that the response spectrum of the input earthquake is closest to the response spectrum of the accelerogram number 15 in the training set. The second column shows the error between the response spectrum of the neural network generated accelerogram and the response spectra of the 30 accelerograms in the training set. The smallest entry is again at the accelerogram number 15, which indicates that the neural network has generated an accelerogram which is very close to (or almost the same as) the 1979 Imperial Valley Earthquake, El Centro array #5 accelerogram. The last two columns show the results for the case shown in Figure 9 when the input response spectrum is the Sylmar record from 1994 Northridge earthquake. Again, it can be seen that the input response spectrum is closest to the response spectrum of the accelerogram number 7 in the training set, which is Pacoima dam accelerogram from 1971 San Fernando earthquake, and that the neural network has generated an accelerogram very close to the accelerogram number 7 in the training set.

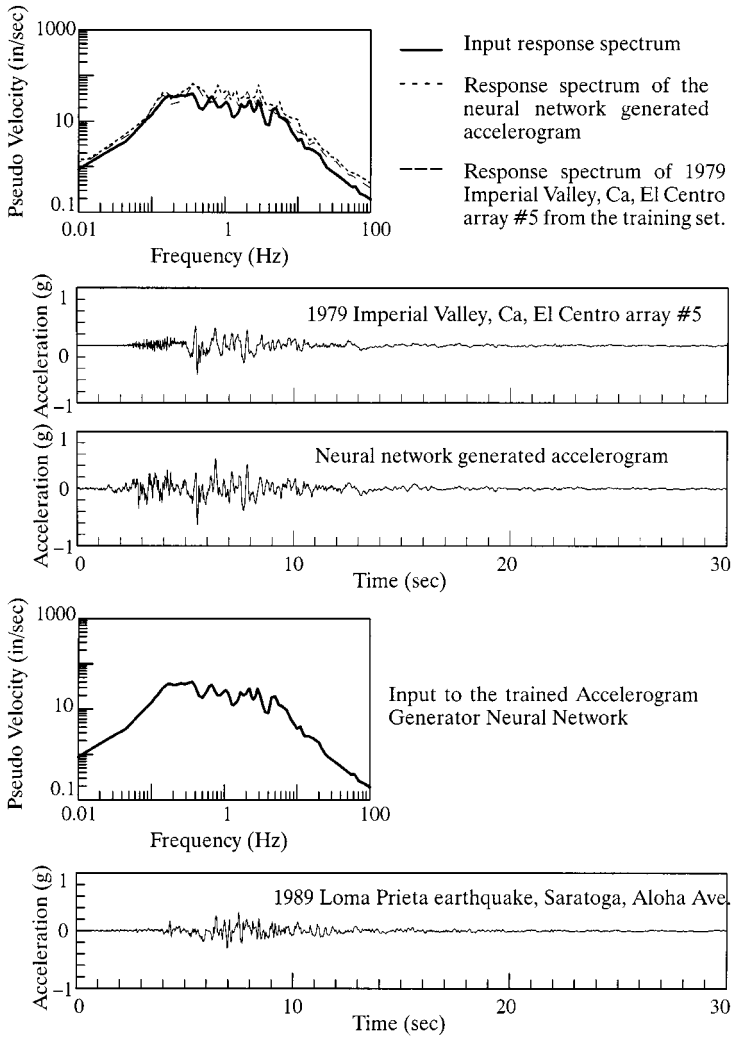

Figure 8. Test of the trained accelerogram generator neural network with a novel response spectrum from an accelerogram which was not part of the training cases. The generated accelerogram is similar to one of the accelerograms in the training set

In case there are no earthquake records in the neural network's training set which have a response spectrum close to the input response spectrum, the trained neural network synthesizes a reasonable looking accelerogram from its training set. Figures 10 and 11 show two such cases. In Figure 10 the input to the neural network is the response spectrum of Newhall record of 1994 Northridge earthquake, and the generated accelerogram is shown in the upper part of the figure. The generated accelerogram does not closely resemble the Northridge earthquake record. However, it is a plausible looking accelerogram and its response spectrum is very close to the input response spectrum. Similarly, Figure 11 shows the case when the input to the neural network is the Corralitos record of 1989 Loma Prieta earthquake. From these two examples it is reasonable to conclude that the trained neural network is capable of generating accelerograms for any novel response spectra in such a way that the response spectrum of the generated accelerogram is close to the input response spectrum. Moreover, the generated accelerograms have the appearance of real recorded

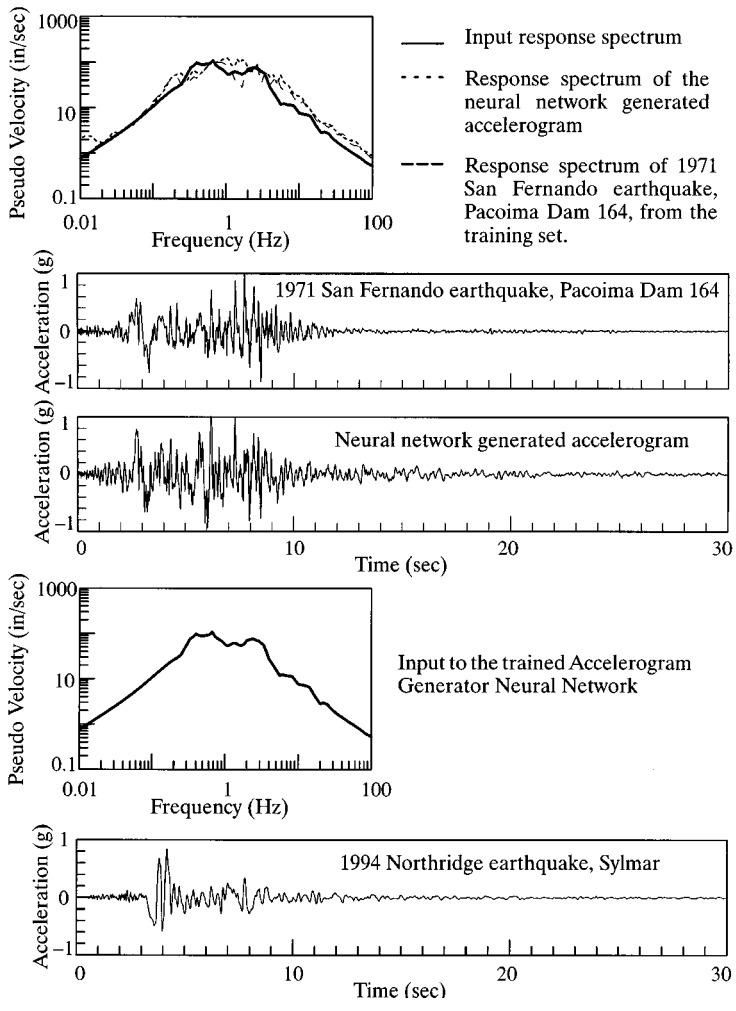

Figure 9. Test of the trained accelerogram generator neural network with a novel response spectrum from an accelerogram which was not part of the training cases. The generated accelerogram is similar to one of the accelerograms in the training set

accelerograms, and they have reasonable and realistic envelopes. The generated accelerograms are generalized from the accelerograms in the training set.

4.2. Generation of accelerograms from design spectra

It is interesting to determine whether the trained neural network is capable of generating reasonable looking accelerograms from design spectra, even though it has been trained with actual recorded earthquake accelerograms. In Figure 12, the trained neural network is provided with a design response spectrum as input, and the generated accelerogram and its response spectrum are shown in the upper part of the figure. The generated accelerogram is a plausible accelerogram with similar characteristics as those in the training set and its response spectrum is very close to the input design spectrum. This is a useful property of the neural

Table II. Comparison of the response spectra of the input and output of AGNN and response spectra of the training set

	1989 Loma Prieta earthquake, Saratoga, Aloha Ave.		1994 Northridge earthquake, Sylmar	
	E_{it}^{*}	$E_{ot}^{\dagger}$	E_{it}	E_{ot}
1	0·0380	0·0386	0·0471	0·0666
2	0·0332	0·0432	0·0372	0·0630
3	0·0951	0·1328	0·1360	0·1682
4	0·0664	0·0905	0·0935	0·1236
5	0·0937	0·1346	0·1378	0·1701
6	0·1234	0·1623	0·1679	0·1988
7	0·0528	0·0325	**0·0234**	**0·0156**
8	0·0818	0·0638	0·0500	0·0544
9	0·0647	0·0885	0·0887	0·1181
10	0·0430	0·0622	0·0649	0·0932
11	0·0623	0·0866	0·0885	0·1184
12	0·0583	0·0413	0·0350	0·0473
13	0·0398	0·0669	0·0703	0·0981
14	0·0320	0·0394	0·0410	0·0670
15	**0·0266**	**0·0157**	0·0335	0·0529
16	0·0376	0·0277	0·0477	0·0627
17	0·0308	0·0297	0·0414	0·0580
18	0·0272	0·0232	0·0292	0·0499
19	0·0455	0·0506	0·0484	0·0751
20	0·0668	0·0672	0·0708	0·0960
21	0·0898	0·0884	0·0884	0·1144
22	0·1123	0·1534	0·1568	0·1898
23	0·1146	0·1286	0·1326	0·1622
24	0·0890	0·0789	0·0724	0·0859
25	0·0921	0·1286	0·1318	0·1638
26	0·0751	0·1096	0·1127	0·1438
27	0·0690	0·0775	0·0813	0·1095
28	0·0823	0·0773	0·0729	0·0996
29	0·0813	0·0758	0·0720	0·1001
30	0·1018	0·1251	0·1252	0·1570

* The response spectrum error of the input of AGNN and the training set
† The response spectrum error of the output of AGNN and the training set

network based methodology, in that it will enable generation of accelerograms compatible with any specified design spectra. The generated accelerograms can then be used in time history analysis of linear and nonlinear structures. This part of the study is performed with the full understanding that design spectra are usually the envelope of the response spectra of several accelerograms or their mean plus some multiple of standard deviation, not the response spectrum of single accelerogram. In the future, we intend to extend the proposed methodology, so that a design spectrum will produce a number of generated accelerograms, the envelope of whose response spectra is close to the input design spectra.

5. CONCLUDING REMARKS

A neural-network-based methodology has been proposed in this paper for generating artificial accelerograms from the pseudo-velocity response spectra. First a replicator neural network is trained to learn the data

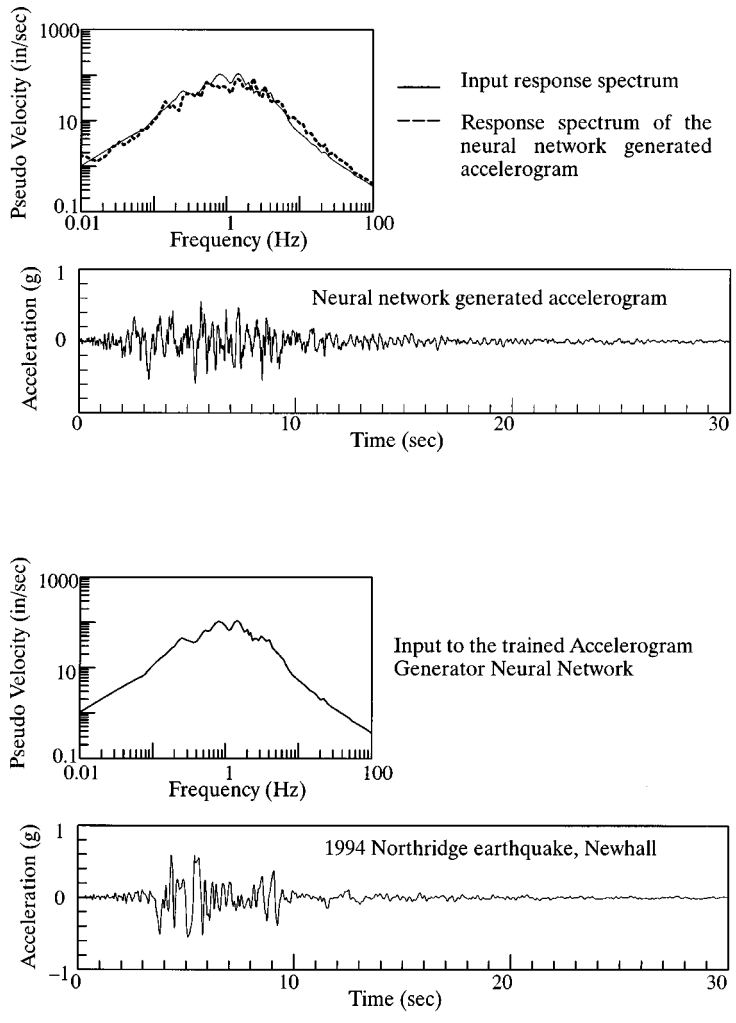

Figure 10. Test of the trained accelerogram generator neural network with a novel response spectrum from an accelerogram which was not part of the training cases. The generated accelerogram is not similar to any of the accelerograms in the training set

compression from FFT of the accelerogram to the Compressed FFT. This is combined with another neural network which is trained to learn to associate the pseudo-velocity response spectra with the Compressed FFT. In an illustrative example, the proposed methodology was applied to a sample of 30 recorded earthquake accelerograms. In testing the trained neural networks, it was found that, when given a pseudo-velocity response spectrum as input, the proposed method either generates an accelerograms very similar to one from its training set, one which has a pseudo-velocity response spectrum close to the input, or it synthesizes a new and realistic looking accelerogram. The proposed methodology was also tested by using design spectra as input and generating accelerograms compatible with those spectra.

More research is needed to train these neural networks with larger number of accelerograms and to further study their properties. A possible approach will be to develop multiple neural networks, and to train each with a group of accelerograms sharing certain properties, such as duration, distance from source, source

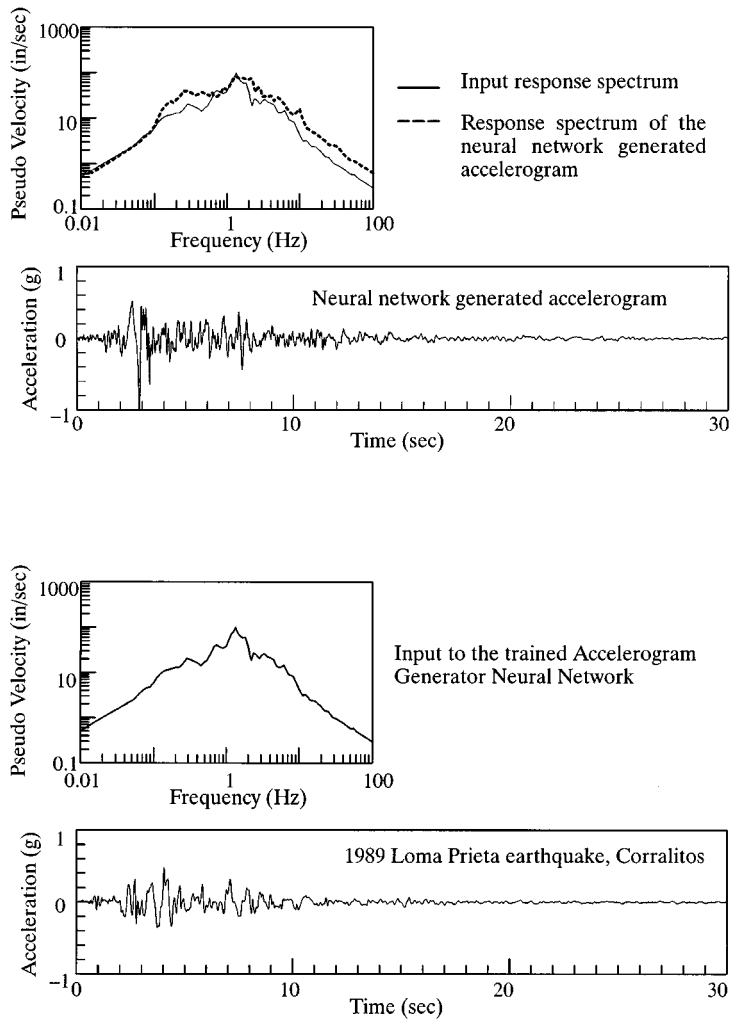

Figure 11. Test of the trained accelerogram generator neural network with a novel response spectrum from an accelerogram which was not part of the training cases. The generated accelerogram is not similar to any of the accelerograms in the training set

characterstics, site characteristics and so on. In this way, it may be possible to generate accelerograms with specified characteristics. Moreover, the proposed methodology offers a systematic way of processing and utilizing the increasing number of the earthquake accelerograms being recorded. As more recorded accelerograms become available they can be used to retrain the neural networks in the proposed methodology.

ACKNOWLEDGEMENTS

The authors wish to acknowledge the valuable contributions of Professor Y.-K. Wen and Dr. Xiping Wu, who participated in discussions during the development of the proposed methodology and gave helpful advice.

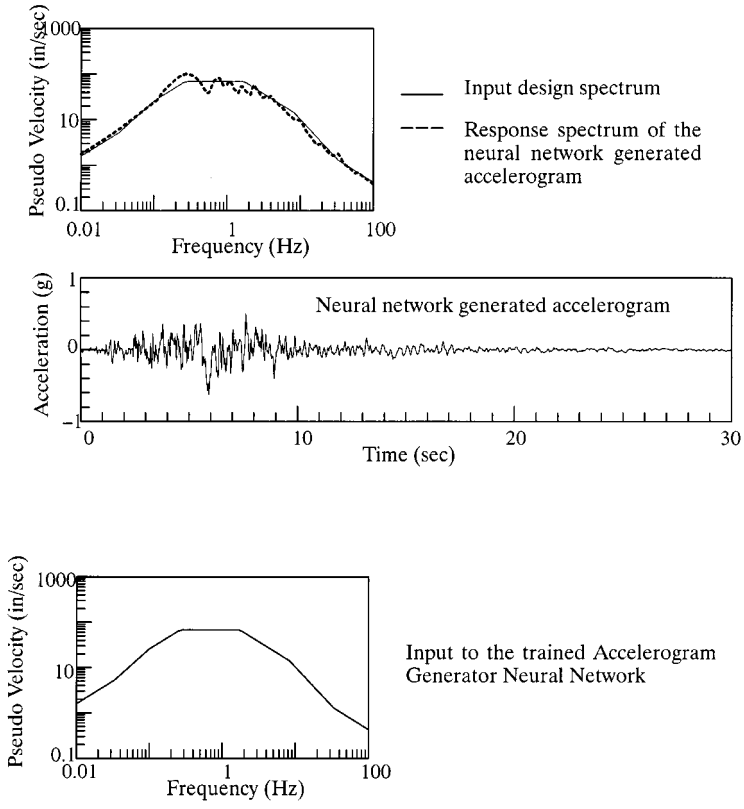

Figure 12. Testing of the trained accelerogram generator neural networks with a design response spectrum

REFERENCES

1. G. W. Housner and P. C. Jennings, 'Generation of artificial earthquakes', *J. Engng. Mech. Div. Proc. ASCE* **90** (EM1), 113–150 (1964).
2. M. Shinozuka, and Y. Sato, 'Simulation of nonstationary random process', *J. Engng. Mech. Div. Proc. ASCE* **93** (EM1), 11–40 (1967).
3. N.-C. Tsai, 'Spectrum-compatible motions for design purposes', *J. Engng. Mech. Div. Proc. ASCE* **98** (EM2), 345–356 (1972).
4. G. R. Saragoni and G. C. Hart, 'Simulation of artificial earthquakes', *Earthquake Engng. Struct. Dyn.* **2**, 249–267 (1974).
5. M. K. Kaul, 'Spectrum-consistent time-history generation', *J. Engng. Mech. Div. Proc. ASCE* **104** (EM4), 781–788 (1978).
6. S. Levy and J. P. D. Wilkinson, 'Generation of artificial time histories, rich in all frequencies, from given response spectra', *Nuclear Engng. Des.* **38**, 241–251 (1976).
7. R. N. Iyengar and P. N. Rao, 'Generation of spectrum compatible accelerograms', *Earthquake Engng. Struct. Dyn.* **7**, 253–263 (1979).
8. H. L. Wong and M. D. Trifunac, 'Generation of artificial strong motion accelerograms', *Earthquake Engng. Struct. Dyn.* **7**, 509–527 (1979).
9. N. W. Polhemus and A. S. Cakmak, 'Simulation of earthquake ground motions using autoregressive moving average (ARMA) Model', *Earthquake Engng. Struct. Dyn.* **19**, 343–354 (1981).
10. M. R. Khan, 'Improved method of generation of artificial time-history, rich in all frequencies, from floor spectra', *Earthquake Engng. Struct. Dyn.* **15**, 985–992 (1987).
11. M. Kimura and M. Izumi, 'A method of artificial generation of earthquake ground motion', *Earthquake Engng. Struct. Dyn.* **18**, 867–874 (1989).
12. P. D. Spanos and M. P. Mignolet, 'Simulation of stationary random processes: two-stage MA to ARMA approach', *J. Engng. Mech.* **116** (3), 620–641 (1990).
13. K. R. Collins, D. A. Foutch and Y. K. Wen, 'Investigation of alternative seismic design procedures for standard buildings', *Civil Engineering Studies, Structural Research Series No. 600*, University of Illinois, Urbana, 1995, pp. 122–132.
14. R. A. W. Haddon, 'Use of empirical green's function, spectral ratios, and kinematic source models for simulating strong ground motion', *Bull. Seism. Soc. Am.* **86**(3), 597–615 (1996).

15. F. Sabetta and A. Pugliese, 'Estimation of response spectra and simulation of nonstationary earthquake ground motions', *Bull. Seism. Soc. Am.* **86**(2), 337–352 (1996).
16. J. Hertz, A. Krogh and R. G. Palmer, *Introduction to the Theory of Neural Computations*, Addison-Wesley, Redwood City, CA, 1991.
17. S. E. Fahlman, 'Faster-learning variations on back-propagation: an empirical study', in *Proceedings of 1988 Connectionist Models Summer School*, Morgan Kaufmann Los Altos CA, 1988, pp. 38–51.
18. A. Cichocki and R. Unbehauen, *Neural Networks for Optimization and Signal Processing*, Wiley, Chichester, 1993.
19. J. Ghaboussi, J. H. Garret Jr. and X. Wu, 'Material modeling with neural networks', *Proc. Int. Conf. on Numerical Methods in Engineering*: *Theory and Applications*, *Swansea*, U.K., 1990, pp. 701–717.
20. J. Ghaboussi, J. H. Garret Jr. and X. Wu, 'Knowledge-based modeling of material behavior wth neural networks', *J. Engng. Mech. Div. ASCE* **117**(1), 132–153 (1991).
21. J. Ghaboussi, 'An overview of the potential applications of neural networks in civil engineering', in *Proc. ASCE Structures Congr. '93*, Irvine, California, 1993.
22. J. Ghaboussi, 'Some applications of neural networks in structural engineering', in *Proc. Structures Congr.*, ASCE, Atlanta, GA, 1994.
23. J. Ghaboussi, M. R. Banan and R. L. Florom, 'Application of neural networks in acoustic wayside fault detection in railway engineering', in *Proc. World Congr. on Railway Research*, Paris, France, 1994.
24. X. Wu, 'Neural network-based material modeling', *Ph.D. Thesis*, University of Illinois at Urbana-Champaign, Urbana, IL.
25. X. Wu and J. Ghaboussi, 'Neural network based material modeling', *Report No. SRS-599*, Department of Civil Engineering, Univ. of Illinois, Urbana, IL.
26. A. Joghataie, J. Ghaboussi and X. Wu, 'Learning and architecture determination through automatic node generation', *Int. Conf. on Artificial Neural Networks in Engineering*, ANNIE '95, St. Louis, Missouri, 1995.
27. J. Ghaboussi, D. A. Pecknold, M. Zhang and R. Haj-Ali, 'Autoprogressive training of neural network constitutive models', *Int. J. Numer. Meth. Engng.* (to appear).
28. J. Ghaboussi and D. E. Sidarta, 'New method of material modeling using neural networks', *Proc. Int. Conf. on Numerical Models in Geomechanics*, Montreal, July 1997.
29. T. Kohonen *et al.*, 'A principle of neural associative memory', *J. Neurosci.* **2**, 1065 (1976).
30. D. H. Ackley, G. E. Hinton and T. J. Sejnowski, 'A learning algorithm for boltzmann machines', *J. Cognitive Sci.* **9**, 147 (1985).
31. G. W. Cottrell, P. Munro and D. Ziper, 'Learning internal representations from gray-scale images: an example of extensional programming', in *Proc. 9th Annual Conf., Cognitive Science Society*, Seattle, WA, 1987.
32. R. Hecht-Nielsen, 'Replicator neural networks for universal optimal source coding', *Science* **269**, 1860–1863 (1995).
33. R. Hecht-Nielsen, 'Data manifolds, natural coordinates, replicator neural networks, and optimal source coding', ICONIP-96, 1996.
34. J. Ghaboussi and D. Sidarata, 'A new nested adaptive neural network for modeling of constitutive behavior of materials', *Int. J. Comput. Geotech.* (to appear).

